

# Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/KR04/003079

International filing date: 26 November 2004 (26.11.2004)

Document type: Certified copy of priority document

Document details: Country/Office: KR  
Number: 10-2003-0085769  
Filing date: 28 November 2003 (28.11.2003)

Date of receipt at the International Bureau: 03 January 2005 (03.01.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland  
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse



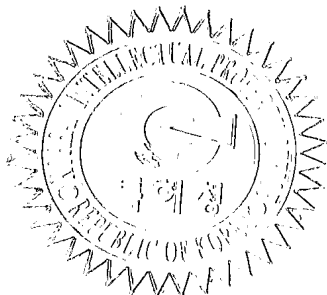
별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원 번호 : 10-2003-0085769  
Application Number

출원 년 월 일 : 2003년 11월 28일  
Date of Application NOV 28, 2003

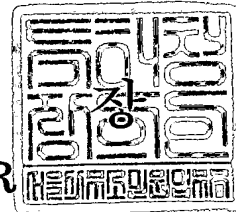
출원인 : 삼성전자주식회사  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2004 년 04 월 02 일

특 허 청

COMMISSIONER



## 【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0004
【제출일자】	2003.11.28
【국제특허분류】	H04L
【발명의 명칭】	저밀도 패리티 검사를 위한 에러 정정 방법 및 장치
【발명의 영문명칭】	Error collection method for low density parity check and the apparatus thereof
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	2003-003435-0
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2003-003436-7
【발명자】	
【성명의 국문표기】	김기현
【성명의 영문표기】	KIM,Ki Hyun
【주민등록번호】	691220-1053119
【우편번호】	463-703
【주소】	경기도 성남시 분당구 구미동(무지개마을) 대림아파트 103동 1103호
【국적】	KR
【발명자】	
【성명의 국문표기】	이윤우
【성명의 영문표기】	LEE,Yoon Woo
【주민등록번호】	650525-1177713

【우편번호】	442-470
【주소】	경기도 수원시 팔달구 영통동 현대아파트 725동 503호
【국적】	KR
【발명자】	
【성명의 국문표기】	김현정
【성명의 영문표기】	KIM,Hyun Jung
【주민등록번호】	731110-2011738
【우편번호】	441-390
【주소】	경기도 수원시 권선구 권선동 삼천리2차아파트 102-1108
【국적】	KR
【취지】	특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인 이영필 (인) 대리인 이해영 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	2 면 2,000 원
【우선권주장료】	0 건 0 원
【심사청구료】	0 항 0 원
【합계】	31,000 원
【첨부서류】	1. 요약서·명세서(도면)_1통

【요약서】

【요약】

본 발명은 저밀도 패리티 검사(LDPC)를 이용한 복호화 장치에 있어서 복호화된 이진 신호의 에러 여부를 판정하고 에러 존재시에는 에러를 정정하는 에러 정정 방법 및 장치에 관한 것이다. 본 발명은, 저밀도 패리티 검사 행렬( $m \times n$ )와 코드 워드 벡터( $n \times 1$ )를 XOR 연산 및 모듈러 2 연산하여 결과 행렬( $m \times 1$ )를 생성하고, 상기 결과 행렬에 기초하여 상기 코드 워드 벡터의 복호화 성공 여부를 판정하는 복호화 성공 여부 검사 단계; 및 상기 복호화가 실패로 판정된 경우, 상기 저밀도 패리티 검사 행렬, 상기 코드 워드 벡터 및 상기 결과 행렬의 구성 성분의 상관 관계에 기초하여 상기 코드 워드 벡터내에서 에러가 발생한 코드 워드 비트를 검출하는 단계를 포함한다. 본 발명에 의하면 저밀도 패리티 검사를 이용한 복호화 장치에서 적은 수의 에러에 의해 전체 블록이 복호 실패로 판정되는 것을 방지하고, 또한 1 비트만이 에러로 판정된 경우에 이를 정정할 수 있다.

【대표도】

도 5

**【명세서】****【발명의 명칭】**

저밀도 패리티 검사를 위한 에러 정정 방법 및 장치{Error collection method for low density parity check and the apparatus thereof}

**【도면의 간단한 설명】**

도 1은 본 발명에 따른 LDPC 에러 정정 방법을 나타내는 시간 흐름도.

도 2는 본 발명에 따른 에러 정정 장치를 나타내는 블록도.

도 3은 규칙적 LDPC 복호화에 있어서 각 행렬의 구성 성분들간의 관계를 나타내는 도면.

도 4는 본 발명의 일 실시예에 따른 에러 판정 및 정정 방법의 원리를 나타내는 행렬 구성 성분간의 관계를 나타내는 도면.

도 5는 본 발명의 일 실시예에 따른 에러 판정 및 정정 방법을 나타내는 시간 흐름도이다.

**【발명의 상세한 설명】****【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<6> 본 발명은 에러 정정 방법 및 장치에 관한 것으로서, 보다 상세하게는 저밀도 패리티 검사(LDPC)를 이용한 복호화 장치에 있어서 복호화된 이진 신호의 에러 여부를 판정하고 에러 존재시에는 에러를 정정하는 에러 정정 방법 및 장치에 관한 것이다.

- <7> 무선통신분야 또는 광 기록재생 분야 등에서 사용되는 에러 정정 부호화 및 복호화 기술의 한 가지로서 저밀도 패리티 검사(Low Density Parity Check, 이하 LDPC 라 한다) 부호화 및 복호화 방법이 있다. LDPC 부호화는 패리티 검사 행렬을 이용하여 패리티 정보를 생성하는 과정을 포함한다. 이 때 패리티 검사 행렬의 구성 성분 대부분은 0 d고 극히 일부 구성 성분만이 1 이다.
- <8> LDPC 부호화는 규칙적 LDPC 부호화(regular LDPC) 와 불규칙적 LDPC 부호화(irregular LDPC)로 나뉜다. 규칙적 LDPC 는 부호화 및 복호화에 사용되는 패리티 검사 행렬이 포함하는 구성 성분 1 의 개수가 행(row) 및 열(column)마다 동일한 개수인 경우이고, 불규칙적 LDPC 는 그렇지 않은 경우이다.
- <9> LDPC 부호화는 이하 수학적 식 1로 표현될 수 있다.
- <10> 【수학적 식 1】  $H \cdot C_e = 0$
- <11> H는 패리티 검사 행렬이고 0는 영행렬,  $\cdot$ 는 XOR 연산 및 모듈러 2 연산을 의미한다.  $C_e$  는 코드 워드 벡터로서, 부호화 대상이 되는 코드 워드를 나타내는 열행렬(column matrix)이다. 코드 워드는 x비트의 메시지 워드  $x_1, x_2, \dots, x_x$  과 p비트의 패리티 정보  $p_1, p_2, \dots, p_p$ 로 이루어진다.
- <12> 패리티 정보  $p_1, p_2, \dots, p_p$  는 각 메시지 워드  $x_1, x_2, \dots, x_x$  가 수학적 식 2를 만족하도록 생성된다. 즉 패리티 검사 행렬 H와 행렬  $C_e$  의 구성 성분중 부호화 대상인 메시지 워드의 이진값은 결정되어 있으므로 상기 수학적 식 1에 의해 패리티 정보  $p_i(i=1,2\dots p)$ 를 결정할 수 있다.
- <13> LDPC 복호화는 이하 수학적 식 2로 표현될 수 있다.

<14> 【수학식 2】  $H \cdot C_d = Z$

<15> 여기서  $H$  는 부호화에서 사용한 것과 동일한 패리티 검사 행렬이고,  $C_d$  는 채널을 통과한 후의 코드 워드 벡터이며,  $Z$  는 이 둘 행렬을 모듈러 2 연산함으로서 생성되는 결과 행렬이다. 만약 복호화가 성공적으로 수행되어서 원래의 코드 워드를 복원하였다면 즉  $C_e = C_d$  라면 결과 행렬  $Z$  는 영행렬이 될 것이다. 즉 복호화의 성공 여부는 결과 행렬  $Z$  의 모든 성분이 0 인지 아닌지를 판정함으로써 복호화의 성공 여부를 판단한다.

<16> LDPC 부호화에 대한 보다 상세한 설명은 "Good error correction codes based on very sparse matrices"(D.J.MacKay, IEEE Trans. on Information Theory, vol. 45, no.2, pp.399-431, 1999) 및 "Efficient encoding of low density parity check codes" (T. Richardson, R.Urbanke, IEEE Trans. on Information Theory, vol.47, no.2, pp638-656, 2001))"에 기술되어 있다.

<17> 그러나, 이러한 종래의 LDPC 복호화 방법에 의하면, 복호화된 코드 워드 벡터  $C_d$  중 1비트만 에러가 발생하더라도 결과 행렬  $Z$  는 영행렬이 되지 않기 때문에, 복호화 성공여부는 실패로 판정되게 되어 효율적인 복호화 방법이라 할 수 없다.

#### 【발명이 이루고자 하는 기술적 과제】

<18> 따라서, 본 발명에 따른 LDPC 복호화 방법 및 장치는 저밀도 패리티 검사를 이용한 복호화 장치에서 적은 수의 에러에 의해 전체 블록이 복호 실패로 판정되는 것을 막고, 또한 1비트만이 에러로 판정된 경우에 이를 정정할 수 있는 저밀도 패리티 검사 에러 정정 방법 및 장치에 관한 것이다.



## 【발명의 구성 및 작용】

<19>       전술한 과제를 해결하기 위해 본 발명은, 저밀도 패리티 검사 복호화의 에러 정정 방법에 있어서, 저밀도 패리티 검사 행렬( $m \times n$ )와 코드 워드 벡터( $n \times 1$ )를 XOR 연산 및 모듈러 2 연산하여 결과 행렬( $m \times 1$ )를 생성하고, 상기 결과 행렬에 기초하여 상기 코드 워드 벡터의 복호화 성공 여부를 판정하는 복호화 성공 여부 검사 단계; 및 상기 복호화가 실패로 판정된 경우, 상기 저밀도 패리티 검사 행렬, 상기 코드 워드 벡터 및 상기 결과 행렬의 구성 성분의 상관 관계에 기초하여 상기 코드 워드 벡터내에서 에러가 발생한 코드 워드 비트를 검출하는 단계를 포함한다.

<20>       또한 본 발명에 따른 에러 판정 방법은, 저밀도 패리티 검사 행렬( $m \times n$ )와 코드 워드 벡터( $n \times 1$ )를 곱하여 결과 행렬( $m \times 1$ )를 생성하고, 상기 결과 행렬에 기초하여 상기 코드 워드 벡터의 복호화 성공 여부를 판정하는 제 1 복호화 성공 여부 판정 단계; 및 상기 복호화가 성공 여부가 실패로 판정된 경우, 상기 결과 행렬에 존재하는 1의 개수에 기초하여 복호화 성공 여부를 다시 판정하는 제 2 복호화 성공 여부 판정 단계를 포함한다.

<21>       또한 본 발명에 따른 에러 정정 장치는, 저밀도 패리티 검사 복호화의 에러 정정 장치에 있어서, 저밀도 패리티 검사 행렬( $m \times n$ )와 코드 워드 벡터( $n \times 1$ )를 XOR 연산 및 모듈러 2 연산하여 결과 행렬( $m \times 1$ )를 생성하고, 상기 결과 행렬에 기초하여 상기 코드 워드 벡터의 복호화 성공 여부를 판정하는 복호화 성공 여부 검사부; 상기 패리티 검사 행렬내에서 상기 결과 행렬과 동일한 열 벡터를 검색하고, 동일한 열 벡터가 존재하는 경우 상기 동일한 열 벡터의 열 번호를 검출함으로써 에러 위치를 검출하는 에러 위치 검출부를 포함하는 것을 특징으로 하는 에러 정정 장치.

<22>       이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 일 실시예를 상세히 설명한다.

- <23> 도 1은 본 발명에 따른 LDPC 에러 정정 방법을 나타내는 시간 흐름도이다.
- <24> 단계 100에서, 복호화 후의 코드 워드 벡터 C에 대해 복호 성공 여부 검사를 수행한다. 이 동작은 종래의 기술과 동일하게 패리티 검사 행렬 H와 복호화 후의 코드 워드 벡터 C를 XOR 연산 및 모듈러 2 연산한 결과 행렬 Z의 구성성분이 모두 0 인지를 검토함으로써 수행된다.
- <25> 단계 100에서 복호화가 실패하였다고 판단되면 즉 결과 행렬 Z에 하나의 "1"이라도 발견되면, 이때 발생한 에러가 1 비트 에러인지 검사한다(단계 110). 1비트 이기 위한 첫 번째 조건은 결과 행렬 Z에 발생한 "1"의 개수가 패리티 검사 행렬 Z의 하나의 열에 포함된 "1"의 개수와 동일하다는 것이다. 즉 1비트 에러의 첫 번째 조건은 에러의 개수에 관한 것이다. 이는 하나의 코드 워드 비트의 변화는 패리티 검사 행렬의 특정 열에 포함된 "1"의 개수만큼 결과 행렬의 생성에 영향을 미치기 때문이다. 결과적으로 결과 행렬 Z 내에 포함된 "1"의 개수와 패리티 검사 행렬의 하나의 열에 포함된 "1"의 개수와 동일하지 않으면 1 비트 에러가 아니라고 판정한다.
- <26> 단계 110에서 1비트 에러가 아니라고 판단되었으면, 복호기는 복호 실패로 판정하고 복호화 과정을 그대로 종료한다. 하지만 반대의 경우에는 복호화된 코드 워드 벡터 C가 1비트 에러이기 위한 두 번째 조건을 만족하는지 여부가 판단되어야 한다(단계 120). 1 비트 에러이기 위한 두 번째 조건은 결과 행렬 Z와 동일한 열 벡터가 패리티 검사 행렬내에 존재하여야 한다는 것이다. 즉 1 비트 에러의 두 번째 조건은 에러의 위치에 관한 것이다. 이 조건에 대해서는 도 3 및 도 4에 관한 설명에서 상세히 설명한다.

- <27> 단계 110 및 120에서 1 비트 에러이기 위한 두 개의 조건을 모두 만족하였다면, 1 비트 에러 정정이 수행된다(단계 130). 1비트 에러 정정 동작은 단계 120에서 검출한 열 벡터의 열 번호에 해당하는 코드 워드 벡터내의 코드 워드 비트의 이진값을 변환함으로써 수행된다.
- <28> 도 2 는 본 발명에 따른 에러 정정 장치를 나타내는 블록도이다.
- <29> 본 발명에 따른 에러 정정 장치는 복호 성공 여부 검사부(210), 에러 개수 검출부(220), 에러 위치 검출부(230) 및 이진값 변환부(240)를 포함한다. 복호 성공 여부 검사부(210)는 복호화된 입력 이진 신호인 코드 워드 벡터 C (211) 및 패리티 검사 행렬 H 로부터 결과 행렬 Z 를 생성하고, 결과 행렬 Z 의 구성성분이 모두 "0" 인지를 검토함으로써 복호 성공 여부를 판단한다.
- <30> 에러 개수 검출부(210)는 복호 성공 여부가 실패로 판단된 경우, 결과 행렬 Z (221)을 수신하여 결과 행렬에 존재하는 "1" 의 개수를 검출한 후, 검출한 "1" 의 개수가 패리티 검사 행렬 H 의 열 웨이트와 동일한 지를 판단한다. 만약 동일하다면 제 1 조건 충족 신호(231)을 생성하여 에러 위치 검출부(230)로 전송한다. 에러 위치 검출부(230)는 제 1 조건 충족 신호(231)를 에러 개수 검출부로부터 수신한 경우, 패리티 검사 행렬 H 내에 결과 행렬 Z 와 동일한 열벡터가 존재하는지를 검색한다. 만약 존재한다면, 에러 위치 검출부(230)는 제 2 조건 충족 신호(241) 및 검색한 열 벡터의 열 번호 k를 이진값 변환부(240)로 전송한다.
- <31> 이진값 변환부(240)는 에러 위치 검출부(230)로부터 제 2 조건 충족 신호(241)를 수신한 경우, 에러 위치 검출부(230)로부터 수신한 열 벡터의 열 번호 k 와 동일한 번호의 코드 워드 벡터 C 내의 코드 워드 비트의 이진값을 변환함으로써 에러를 정정한다.

- <32> 도 3 은 규칙적 LDPC 복호화에 있어서 각 행렬의 구성 성분들간의 관계를 나타내는 도면이다.
- <33> 저밀도 패리티 검사 행렬  $H$  는 구성 성분  $h_{11}$  내지  $h_{mn}$  을 가지는  $m \times n$  행렬이다. 규칙적 LDPC 부호화에서, 각 행  $R_1, R_2, R_3, \dots, R_m$  에 있는 구성 성분 1 의 개수는 동일하고, 또한 각 열  $C_1, C_2, \dots, C_n$  에 존재하는 구성 성분 1 의 개수는 동일하다. 이 때 각 행  $R_1, R_2, R_3, \dots, R_m$  에 존재하는 1 의 개수를 행 웨이트(Row weight)라고 하고, 각 열에 존재하는 1 의 개수를 열 웨이트(column weight)라고 한다. 일반적인 규칙적 LDPC 부호화에서, 행 웨이트는 3 이고, 열 웨이트는 보통 행 웨이트의 3 배수인 9 이다.
- <34> 규칙적 LDPC 복호화에서, 패리티 검사 행렬  $H$  의 각 행  $R_1, R_2, R_3, \dots, R_m$  은 모두 상이하고, 각 열  $C_1, C_2, C_3, \dots, C_n$  또한 모두 상이하다. 즉  $R_1 \neq R_2 \neq R_3, \dots \neq R_m$  이고 또한  $C_1 \neq C_2 \neq C_3, \dots \neq C_n$  이다. 이러한 성질은 후술할 에러 정정에 이용된다.
- <35> 복호화된 코드 워드 벡터  $C'$  는 정보 비트와 패리티 비트를 포함하는 코드 워드 비트  $x_1, x_2, \dots, x_n$  를 포함한다. 패리티 검사 행렬  $H$  및 코드 워드 벡터  $C'$  는 모듈러 연산됨으로써 결과 행렬  $Z$  를 생성한다.
- <36> 각 행렬의 성분들 간의 관계를 정리하면 이하 수학식 3으로 요약된다.
- <37> 【수학식 3】  $\text{mod } 2[h_{11}x_1 + h_{12}x_2 + \dots h_{1n}x_n] = z_1$
- <38>  $\text{mod } 2[h_{21}x_1 + h_{22}x_2 + \dots h_{2n}x_n] = z_2$
- <39> ...
- <40>  $\text{mod } 2[h_{m1}x_1 + h_{m2}x_2 + \dots h_{mn}x_n] = z_m$
- <41> 이 때  $z_1, z_2, \dots, z_m$  중 어느 하나라도 1 이라면 LDPC 복호화는 실패로 판정된다.

- <42> 도 4 는 본 발명의 일 실시예에 따른 에러 판정 및 정정 방법의 원리를 나타내는 행렬 구성 성분간의 관계를 나타내는 도면이다.
- <43> 도 4에서, 열 웨이트(column weight, CW) 는 3 이고 패리티 검사 행렬 H 는 10\*20 의 행렬이다. 코드 워드 비트  $x_1, x_2, \dots, x_{20}$  는 복호화된 코드 워드 비트를 나타낸다. 결과 행렬 Z 는 구성 성분이 10개인 열 벡터이다.
- <44> 이 실시예에서 결과 행렬 Z 는 3, 7, 10번째 구성 성분이 1 이다. 이 결과 행렬은 LDPC 복호화가 실패하였음을 나타낸다. 이제 결과 행렬 Z 의 각 구성 성분들이 패리티 검사 행렬 및 코드 워드 벡터의 어떤 성분으로부터 발생되었으며 이것이 에러의 원인에 어떠한 영향을 주는 지 검토한다. 이는 수학적 3을 검토함으로써 알 수 있다.
- <45> 결과 행렬 Z 의 첫 번째 "1" (411)은 패리티 검사 행렬 H 의 세 번째 행 R3 과 코드 워드 열 벡터의 모듈러 연산으로부터 생성된다. 결과 행렬 Z 의 두번째 분 "1" (412)은 패리티 검사 행렬 H 의 일곱 번째 행 R7 과 코드 워드 열 벡터의 모듈러 연산으로부터 생성된다. 마찬가지로, 결과 행렬 Z 의 세 번째 "1"(411)은 패리티 검사 행렬 H 의 열번째 행 R10 과 코드 워드 열 벡터의 모듈러 연산으로부터 생성된다. 만약 복호화 후의 코드 워드 벡터 C' 가 원래의 코드 워드 벡터 C 와 동일하다면, 결과 행렬 Z 에는 "1" 이 나타나지 않았을 것이다. 그러나 1 이 나타난 것으로 보아 코드 워드 벡터 C' 의 코드 워드 비트들 중에 어느 하나 혹은 그 이상(아직 모른다)의 이진값이 변환되었음을 예측할 수 있다. 본 발명은 이러한 경우중 하나의 코드 워드 비트 만이 변화된 경우 즉 1비트 에러가 발생한 경우에 에러 발생 위치를 예측한다.
- <46> 코드 워드 벡터중 1비트에만 에러가 발생하였고, 이 에러가 발생한 위치가 코드 워드 벡터 C' 의 10번째 코드 워드 비트  $x_{10}$  이라고 가정하자. 수학적 3을 살펴보면, 결과 행렬을 생성하는데 있어서 코드 워드 비트 x

10 은 패리티 검사 행렬의 10번째 열 C10 의 각 비트들과 모듈러 연산이 수행됨을 알 수 있다. 그러나 코드 워드 비트  $x_{10}$  이 열 C10 의 모든 비트와 모듈러 연산이 수행되지는 않는다. 0 인 구성성분도 있기 때문이다. 즉 코드 워드 비트  $x_{10}$  은 C10 의 구성 성분중 1 이 존재하는 위치에서만 모듈러 연산이 수행되고, 그 결과 이 위치에서만 결과 행렬의 구성성분의 생성에 영향을 미친다. 이 실시예에서, C10 내에서 구성 성분이 1 인 위치는 3,7,10 번째 이므로, 결과적으로 결과 행렬 또한 3,7,10 번째 위치에서만 1 이 나타난 것이다.

<47> 결과적으로, 만약 1비트 에러 만이 발생한 경우라면, 결과 행렬에서 1 이 나타난 위치와 동일한 위치에 1을 가지는 패리티 검사 행렬 H 내의 열 번호는, 코드 워드 벡터내에서 에러가 발생한 코드 워드 비트의 번호와 동일함을 알 수 있다. 다만 이러한 명제는 H 가 규칙적 LDPC 의 패리티 검사 행렬이고, 코드 워드 중 1비트에서만 에러가 발생하였다는 가정에서만 성립된다. 만약 1비트 에러가 아니라면, 결과 행렬의 구성성분의 생성에 있어서 수 개의 코드 워드들이 영향을 미치므로 패리티 검사 행렬의 열과 결과 행렬의 구성 성분의 생성은 일대일 관계가 아니기 때문에, 우리는 어느 코드 워드 비트의 변화가 결과 행렬의 생성에 영향을 미쳤는지 알 수 없다.

<48> 행렬의 구성 성분들이 가지는 위의 성질을 요약하면, 결과 행렬을 검토함으로써 코드 워드의 몇 번째 비트에서 에러가 발생하였는지 알 수 있다. 이는 다음 단계에 의해 수행된다.

<49> 첫째, 결과 행렬에서 1 이 나타난 비트 번호를 검출한다. 도 4 의 실시예에서 결과 행렬 내 구성 성분이 1 을 가지는 비트 번호는 3,7,10 이다.

<50> 둘째, 첫 번째 단계에서 검출된 비트 위치에 1을 가지는 패리티 검사 행렬의 열 C를 찾는다. 규칙적 LDPC 에서는 패리티 검사 행렬의 열은 모두 상이하므로 이러한 열은 오직 하나이다. 도 4 의 실시예에서 이 열은 C10이다.

- <51> 셋째, 두 번째 단계에서 구한 열의 열 번호와 동일한 번호의 코드 워드 비트가 바로 에러가 발생한 코드 워드 비트의 번호이다. 도 4의 실시예에서 이 코드 워드 비트는  $x_{10}$  이다.
- <52> 전술한 바와 같이, 본 발명은 1 비트 에러가 발생하였다는 사실과 패리티 검사 행렬의 각 열은 동일한 열 웨이트를 가져야 한다는 사실을 전제로 한다. 따라서 위 첫 번째 단계에서 결과 행렬에서 1 이 나타난 비트의 개수와 열 웨이트가 동일하지 않다면 (예를 들어 본 실시예에서 결과 행렬에서 1 이 나타난 비트 수가 4개라면) 본 발명에 따른 방법을 적용할 수 없다.
- <53> 광 디스크 시스템에서 사용하는 LDPC 패리티 검사 행렬은 일반적으로  $m = 1000$  내지 10000,  $n = 3000$  내지 30000 이다. 이에 대해 DVD 의 일반적인 비트 에러율이  $10^{-12}$  임을 감안하면, 하나의 코드 워드 벡터 (3000 내지 30000 개의 비트)에 대해 2비트 이상의 에러가 발생할 확률은 거의 없다. 따라서 일반적인 광 디스크 시스템에서는 1비트 에러의 정정만으로도 매우 큰 에러 정정 효과를 달성할 수 있다.
- <54> 도 5 는 본 발명의 일 실시예에 따른 에러 판정 및 정정 방법을 나타내는 시간 흐름도이다.
- <55> 단계 510 및 520 는 복호후 코드 워드 벡터 C 의 에러 존재 여부를 검출하는 과정이다. 복호 성공 여부 검사부(210)는 부호화시에 사용된 패리티 검사 행렬 H 와 복호화된 코드 워드 벡터 C를 곱하고 각 성분에 대해 모듈러 2 연산을 함으로써 결과 행렬 Z를 생성한다(단계 510). 그 후 복호 성공 여부 검사부(210)는 결과 행렬 Z 의 구성 성분이 모두 "0" 인지를 검사한다(단계 520). 모듈러 2 연산을 하였으므로 복호화된 코드 워드 벡터 C 에 에러가 발생하지 않았다면 결과 행렬 Z 는 모두 구성 성분이 0 일 것이다. 그러나, 만약 에러가 발생하여 하나의 "1" 이라도 존재한다면 단계 530 으로 진행한다.

<56> 단계 530 및 540 은 코드 워드 벡터 C 에 발생한 에러가 본 발명에 의해 정정 가능한 에러인지 즉 1비트 에러인지를 검토하는 첫 번째 과정이다. 단계 520에서 결과 행렬 Z 의 구성 성분이 하나의 1이라도 포함한다고 판단된 경우, 에러 개수 검출부(220)는 결과 행렬 Z 내에 존재하는 "1" 의 개수를 검출하고(단계 530), 이 코드 워드 벡터 내의 "1" 의 개수가 패리티 검사 행렬 H 의 열 웨이트(CW)와 동일한지 여부를 판단한다(단계 540). 만약 동일하지 않다면, 1비트 에러가 아니므로 본 발명에 의한 에러 정정이 불가능하므로, 복호 실패로 판정한다. 만약 동일하다면, 단계 550 으로 진행한다.

<57> 단계 550 내지 570 은 코드 워드 벡터 C 에 발생한 에러가 본 발명에 의해 정정 가능한 에러인지 즉 1비트 에러인지를 검토하는 두 번째 과정이다. 단계 540에서 코드 워드 벡터 C 내의 "1" 의 개수가 패리티 검사 행렬 H 의 열 웨이트(CW)와 동일하다고 판단되면, 에러 위치 검출부(230)는 패리티 검사 행렬 H 내에 존재하는 열 행렬  $C_1, C_2, \dots, C_n$  각각과 결과 행렬 Z을 비교함으로써(단계 550) 결과 행렬 Z 와 동일한 열 행렬  $C_k$  가 존재하는지를 판단한다(단계 560). 만약 존재하지 않는다면, 1비트 에러가 아닌 제 3 의 요인에 의해 결과 행렬에 "1" 이 발생한 것이므로, 본 발명에 의한 에러 정정을 수행할 수 없기 때문에 복호 실패로 판정된다. 만약 존재한다면, 해당 열 행렬의 열 번호 k 를 추출한다(단계 570).

<58> 이진값 변환부(240)는 단계 570 에 코드 워드 벡터내의 k 번째 코드 워드 비트의 이진값이 0 이라면 1 로 1 이라면 0 으로 변환함으로써(단계 580) 정정후의 코드 워드 벡터 C'를 생성한다. 코드 워드 비트는 0 또는 1 의 값만 가지므로 만약 0 값에 에러가 생겼다면 1 이 에러가 생기기 전의 값이기 때문에 이러한 이진값 변환은 에러를 정정한다.



<59> 단계 590 및 단계 600 에서는 복호 성공 여부 검사부(210)는 정정후의 코드 워드 벡터 C' 에 대해 단계 510 및 520 에서와 동일한 방법으로 복호 성공 여부를 검사함으로써 정정이 수행되었는지를 다시 한번 확인한다.

<60> 이제까지 본 발명에 대하여 그 바람직한 실시예들을 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

#### 【발명의 효과】

<61> 전술한 바와 같이 본 발명에 따른 LDPC 복호화 방법 및 장치에 의하면, 저밀도 패리티 검사를 이용한 복호화 장치에서 적은 수의 에러에 의해 전체 블록이 복호 실패로 판정되는 것을 방지하고, 또한 1 비트만이 에러로 판정된 경우에 이를 정정할 수 있다.

**【특허청구범위】****【청구항 1】**

저밀도 패리티 검사 복호화의 에러 정정 방법에 있어서,

저밀도 패리티 검사 행렬( $m \times n$ )와 코드 워드 벡터( $n \times 1$ )를 XOR 연산 및 모듈러 2 연산하여 결과 행렬( $m \times 1$ )를 생성하고, 상기 결과 행렬에 기초하여 상기 코드 워드 벡터의 복호화 성공 여부를 판정하는 복호화 성공 여부 검사 단계;

상기 복호화가 실패로 판정된 경우, 상기 저밀도 패리티 검사 행렬, 상기 코드 워드 벡터 및 상기 결과 행렬의 구성 성분의 상관 관계에 기초하여 상기 코드 워드 벡터내에서 에러가 발생한 코드 워드 비트를 검출하는 단계; 및

상기 검출된 코드 워드 비트의 이진값을 변환함으로써 에러를 정정하는 이진값 변환 단계

를 포함하는 것을 특징으로 하는 에러 정정 방법.

**【청구항 2】**

제 1 항에 있어서, 상기 에러가 발생한 코드 워드 비트를 검출하는 단계는,

상기 패리티 검사 행렬에 상기 결과 행렬과 동일한 열 벡터가 존재하는지를 검색하는 단계;

만약 결과 행렬과 동일한 열 벡터가 존재한다면, 상기 검색된 열 벡터의 열 번호에 해당하는 번호의 코드 워드 벡터의 코드 워드 비트를 에러 발생 코드 워드 비트로서 검출하는 단계를 포함하는 것을 특징으로 하는 에러 정정 방법.

**【청구항 3】**

제 1 항에 있어서, 상기 에러가 발생한 코드 워드 비트를 검출하는 단계는,

만약 결과 행렬과 동일한 열 벡터가 존재하지 않는다면, 복호 실패로 판정하고 에러 정정 과정을 종료하는 단계를 더 포함하는 것을 특징으로 하는 에러 정정 방법.

**【청구항 4】**

제 1 항에 있어서, 상기 결과 행렬에 존재하는 1의 개수에 기초하여 1 비트 에러 발생 개수를 검출하는 단계를 더 포함하는 에러 정정 방법.

**【청구항 5】**

제 4 항에 있어서, 상기 에러 발생 개수를 검출하는 단계는,

상기 결과 행렬의 1의 개수와 상기 패리티 검사 행렬의 열 웨이트가 동일한 지를 비교하는 단계; 및

상기 결과 행렬의 1의 개수와 상기 패리티 검사 행렬이 열 웨이트가 동일하지 않다면, 복호 실패로 판정하고 에러 정정 과정을 종료하는 단계를 포함하는 것을 특징으로 하는 에러 정정 방법.

**【청구항 6】**

제 3 항에 있어서, 상기 저밀도 패리티 검사 행렬( $m \times n$ )와 상기 코드 워드 비트의 이진값이 변환된 코드 워드 벡터( $n \times 1$ )를 XOR 연산 및 모듈러 2 연산함으로써 새로운 결과 행렬( $m \times 1$ )을 생성하고, 상기 새로운 결과 행렬에 기초하여 상기 코드 워드 벡터의 복호화 성공 여부를 판정하는 복호화 성공 여부 검사 단계를 더 포함하는 것을 특징으로 하는 에러 정정 방법.

**【청구항 7】**

저밀도 패리티 검사에 의한 에러 판정 방법에 있어서,

저밀도 패리티 검사 행렬( $m \times n$ )와 코드 워드 벡터( $n \times 1$ )를 곱하여 결과 행렬( $m \times 1$ )를 생성하고, 상기 결과 행렬에 기초하여 상기 코드 워드 벡터의 복호화 성공 여부를 판정하는 제 1 복호화 성공 여부 판정 단계; 및

상기 복호화가 성공 여부가 실패로 판정된 경우, 상기 결과 행렬에 존재하는 1의 개수에 기초하여 복호화 성공 여부를 다시 판정하는 제 2 복호화 성공 여부 판정 단계를 포함하는 에러 판정 방법.

**【청구항 8】**

제 7 항에 있어서, 상기 제 2 복호화 성공 여부 판정 단계는, 상기 결과 행렬내의 1의 개수가 상기 패리티 검사 행렬의 열 웨이트와 동일하지 않은 경우 복호화 실패로 판정하는 것을 특징으로 하는 에러 판정 방법.

**【청구항 9】**

제 7 항에 있어서, 상기 패리티 검사 행렬내에 상기 결과 행렬과 동일한 열 행렬이 존재하는 지를 판단함으로써 복호화 성공 여부를 판정하는 제 3 복호화 성공 여부 판정 단계를 포함하는 에러 판정 방법.

**【청구항 10】**

저밀도 패리티 검사 복호화의 에러 정정 장치에 있어서,

저밀도 패리티 검사 행렬( $m \times n$ )와 코드 워드 벡터( $n \times 1$ )를 XOR 연산 및 모듈러 2 연산하여 결과 행렬( $m \times 1$ )를 생성하고, 상기 결과 행렬에 기초하여 상기 코드 워드 벡터의 복호화 성공 여부를 판정하는 복호화 성공 여부 검사부;

상기 패리티 검사 행렬내에서 상기 결과 행렬과 동일한 열 벡터를 검색하고, 동일한 열 벡터가 존재하는 경우 상기 동일한 열 벡터의 열 번호를 검출함으로써 에러 위치를 검출하는 에러 위치 검출부; 및

상기 검출된 열 번호에 해당하는 번호의 코드 워드 벡터내의 코드 워드 비트의 이진값을 변환함으로써 에러를 정정하는 이진값 변환부

를 포함하는 것을 특징으로 하는 에러 정정 장치.

#### 【청구항 11】

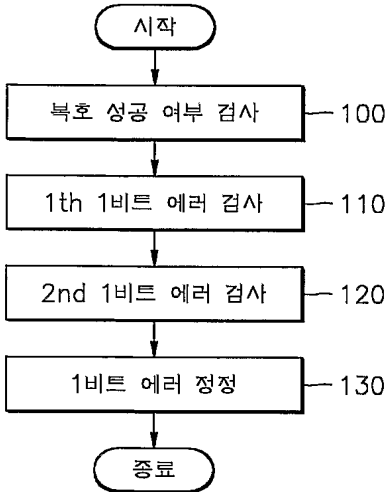
제 10 항에 있어서, 상기 에러 위치 검출부는, 상기 결과 행렬내에서 상기 결과 행렬과 동일한 열 벡터가 존재하지 않는 경우, 복호 실패로 판정하고 에러 정정 과정을 종료하는 것을 특징으로 하는 에러 정정 장치.

#### 【청구항 12】

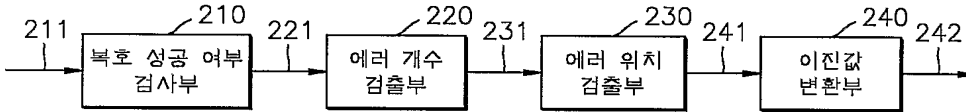
제 10 항에 있어서, 상기 결과 행렬의 1 의 개수와 상기 패리티 검사 행렬의 열 웨이트가 동일한 지를 판단하고, 동일하지 않은 경우 복호 실패로 판정하고 에러 정정 과정을 종료하는 에러 개수 검출부를 더 포함하는 것을 특징으로 하는 에러 정정 장치.

【도면】

【도 1】



【도 2】



【도 3】

$$\begin{matrix} & C_1 & C_2 & \cdots & C_k & \cdots & C_n \\ \begin{matrix} R_1 \\ R_2 \\ \vdots \\ R_j \\ \vdots \\ R_m \end{matrix} & \left[ \begin{array}{cccccc} h_{11} & h_{12} & \cdots & h_{1k} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2k} & \cdots & h_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ h_{j1} & \cdots & & h_{jk} & \cdots & h_{jn} \\ \vdots & \vdots & & \vdots & & \vdots \\ h_{m1} & \cdots & & h_{mk} & \cdots & h_{mn} \end{array} \right] & \begin{matrix} \left[ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{matrix} \right] \\ C \end{matrix} & = & \begin{matrix} \left[ \begin{matrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_m \end{matrix} \right] \\ Z \end{matrix}
 \end{matrix}$$

【도 4】

$$\begin{array}{c}
 R_1 \\
 R_2 \\
 \vdots \\
 R_{10}
 \end{array}
 \begin{array}{c}
 C_1 \quad C_2 \quad \dots \quad C_{10} \quad \dots \quad C_{20} \\
 \left[ \begin{array}{cccccc}
 1 & 0 & & 0 & & 0 \\
 0 & 1 & & 0 & & 0 \\
 1 & 0 & & 1 & & 0 \\
 0 & 1 & & 0 & & 0 \\
 1 & 0 & & 0 & & 0 \\
 0 & 1 & & 0 & & 1 \\
 0 & 0 & & 1 & & 1 \\
 0 & 0 & & 0 & & 1 \\
 0 & 0 & & 0 & & 0 \\
 0 & 0 & & 1 & & 0
 \end{array} \right]
 \end{array}
 \begin{array}{c}
 \\
 \\
 \text{--- 401 ---} \\
 \\
 \text{--- 402 ---} \\
 \\
 \text{--- 403 ---} \\
 \\
 \\
 \end{array}
 \begin{array}{c}
 \left[ \begin{array}{c}
 x_1 \\
 x_2 \\
 \vdots \\
 x_{10} \\
 \vdots \\
 x_{20}
 \end{array} \right]
 =
 \left[ \begin{array}{c}
 0 \\
 0 \\
 1 \\
 0 \\
 0 \\
 0 \\
 1 \\
 0 \\
 0 \\
 1
 \end{array} \right]
 \begin{array}{c}
 \\
 \\
 \text{--- 411 ---} \\
 \\
 \text{--- 412 ---} \\
 \\
 \text{--- 413 ---} \\
 \\
 \\
 \end{array}
 \end{array}$$

【도 5】

